

A STUDY OF THE OPTIMIZATION
OF SOME CHEMICAL PROCESSES

by

CHIU SEN WANG

B.S., National Taiwan University, 1960

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Chemical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1963

Approved by:

Liang-tung Fan

Major Professor

LD
2668
T4
1963
W24
C.2
Document

TABLE OF CONTENTS

	Page
INTRODUCTION	1
THE MAXIMUM PRINCIPLE.	5
APPLICATION TO ONE-DIMENSIONAL SYSTEMS	19
General Equations.	19
Cross-current Extraction Process	23
The Continuous Flow Stirred Tank Reactor with a Single Reaction	38
APPLICATION TO HETEROGENEOUS SYSTEMS	50
ACKNOWLEDGEMENT.	58
BIBLIOGRAPHY	59
NOMENCLATURE	61

INTRODUCTION

The functions of chemical engineers are to develop, design, and/or operate chemical processes to transform economically raw materials into products suitable for use by human beings. Optimization is defined as the design and operation of processes in such a way that the maximum profit may be obtained.

Practically every chemical plant consists of a large number of interconnected process units. To design the whole plant, it is necessary to select the process parameters and the operating variables for each unit. There often are restrictions on the ranges of operating variables, and qualities and quantities of raw materials, or the minimum requirements on the qualities and quantities of the products. The optimization of chemical processes hence is a problem of extrema with a large number of variables, often coupled with inequality constraints. Many methods, e.g., differential calculus, calculus of variations, dynamic programming, and the maximum principle, have been developed to solve the extremal problems. However, most of these methods become less suitable for use as the number of the variables of the function to be extremized increases, or when inequality restrictions are imposed. Dynamic programming and the maximum principle are the two methods derived to attack such multi-variable problems.

It is convenient to define certain terms before introducing the notions of dynamic programming and the maximum principle.

- (1) Stages represent the units of a process, time periods,

or any real or abstract notion, in which certain transformations are taking place.

(2) State variables are those variables which are transformed in the stages.

(3) Control variables are those variables which can be manipulated to achieve desired transformations of state variables in a stage.

(4) Performance equations are those equations which describe the transformations of the state variables in a stage.

(5) A simple system represents several stages connected in series, in which the output from one stage becomes the input to the next stage.

(6) A complex system is a group of stages joined in a form of a branch or loop.

(7) A homogeneous system is a system in which all stages have the same performance equations.

(8) A heterogeneous system is a system in which there are two or more different performance equations.

(9) The objective function of a system is a function or a functional to be maximized or minimized by manipulating the control variables.

(10) The dimension of a system is the number of state variables, other than the objective function, on which the magnitude of the objective function depends.

The underlying idea of dynamic programming, first introduced by Bellman (1) is represented by the principle of optimality, which states, "An optimal policy has the property that whatever

the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision." By means of this principle and the method of interpolation, a multi-stage decision problem is replaced by a sequence of single stage decision problems, for which the solution is drastically simplified. Using the methods of dynamic programming, Aris has solved a variety of problems concerned with chemical reactor design (2) and a cross-current extraction problem (3); Roberts has formulated the catalyst replacement problem (4) and the control of a batch reactor (5); Rudd has investigated the reliability in chemical system design (6) and the optimal use of limited resources (7). A dynamic programming model for countercurrent flow processes has been developed by Dranoff et al (8). Mitten and Nemhauser (9) have applied the techniques of dynamic programming to multi-stage mass transfer and solid separation processes, as well as heterogeneous and complex systems.

While the use of interpolation makes dynamic programming a powerful tool, it also introduces error which is often neither negligible nor easy to estimate. Another difficulty encountered in dynamic programming is its immense computer memory requirement which increases exponentially with the dimensions of the system. In order to circumvent these practical difficulties, a method which avoids the use of interpolation is needed. This is provided by the maximum principle. The optimization technique based on this principle is less well known. It appears, however, to offer many computational advantages over the method of

dynamic programming by overcoming the difficulties mentioned above.

The principal purpose of this thesis is to develop further the method of optimization based on the maximum principle, following the line suggested by Katz (10), (11), (12), into a form particularly suitable in the optimal design of multi-stage chemical processes. Its use will be illustrated in detail by applying it to several concrete examples.

THE MAXIMUM PRINCIPLE

The maximum principle was first developed for continuous processes by the Russian mathematician Pontryagin (13). Recently, Katz (10), (11) has presented a discrete version of Pontryagin's maximum principle for simple homogeneous systems. It will be explicitly shown in this section that Katz's formulation can be extended to heterogeneous systems with product recycle, in which performance equations are different from stage to stage.

An N-stage process is schematically represented in Fig. 1. x_i^{n*} , $i = 1, 2, \dots, S$, are the state variables leaving the n^{th} stage and θ_p^n , $p = 1, 2, \dots, t$, the control variables at the n^{th} stage. Each stage transforms the state variables according to the equations:

$$x_i^n = F_i^n (x_k^{n-1}; \theta_r^n) \quad (1)$$

$$i = 1, 2, \dots, S \quad n = 1, 2, \dots, N$$

Here the notation $F_i^n (x_k^{n-1}; \theta_r^n)$ is a shortened form representing $F_i^n (x_1^{n-1}, x_2^{n-1}, \dots, x_s^{n-1}; \theta_1^n, \theta_2^n, \dots, \theta_t^n)$. It is to be noted that $F_i^1 (x_k^0; \theta_r^1)$, $F_i^2 (x_k^1; \theta_r^2)$, \dots , $F_i^N (x_k^{N-1}; \theta_r^N)$, are

*The superscript, n, indicates the stage number. Exponents will be written with parentheses, such as $(x_1^n)^2$, to distinguish them from superscripts.

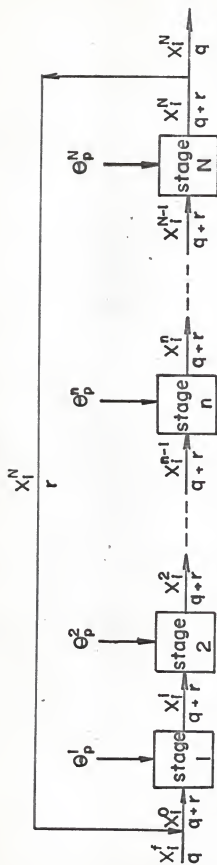


Fig. 1. Schematic representation of multi-stage process with recycle.

different functions of the same variables x_k^{n-1} and θ_r^n ,
 $n = 1, 2, \dots, N$.

The feed enters the system at a rate q , while the recycle stream is fed back at a rate r . The mixing conditions are represented by:

$$x_i^0 = F_i^0 (x_k^f; x_k^N) \quad i = 1, 2, \dots, S \quad (2)$$

where x_k^f are the values of state variables in the feed.

The problem is to find the sequence of θ_r^n , subject to the constraints, $\xi_p^n \leq \theta_p^n \leq \gamma_p^n$; $n = 1, 2, \dots, N$; $p = 1, 2, \dots, t$, which will minimize x_m^N .

Suppose that the minimizing sequence of control actions is found to be $\overline{\theta_r^n}$, and the corresponding state variables $\overline{x_k^n}$, then:

$$\overline{x_i^n} = F_i^n (\overline{x_k^{n-1}}; \overline{\theta_r^n}) \quad (3)$$

$$i = 1, 2, \dots, S$$

$$n = 1, 2, \dots, N$$

and

$$\overline{x_i^0} = F_i^0 (x_k^f, \overline{x_k^N}) \quad i = 1, 2, \dots, S \quad (4)$$

If independent small perturbations of $\overline{\theta_p^n}$ are made at each stage,

$$\theta_p^n = \overline{\theta_p^n} + \epsilon \varphi_p^n \quad (5)$$

$$n = 1, 2, \dots, N;$$

$$p = 1, 2, \dots, t$$

the disturbances will alter the values of $\overline{x_1^n}$ to

$$x_1^n = \overline{x_1^n} + \epsilon y_1^n + O(\epsilon^2) \quad (6)$$

$$n = 0, 1, \dots, N; \quad i = 1, 2, \dots, S$$

satisfying $x_1^n = F_1^n(x_k^{n-1}; \theta_r^n)$.

Combining equations (1), (3), and (6) gives:

$$\begin{aligned} \epsilon y_1^n &= F_1^n(x_k^{n-1}; \theta_r^n) - F_1^n(\overline{x_k^{n-1}}; \overline{\theta_r^n}) + O(\epsilon^2) \\ &= F_1^n(x_k^{n-1}; \theta_r^n) - F_1^n(\overline{x_k^{n-1}}; \theta_r^n) + F_1^n(\overline{x_k^{n-1}}; \theta_r^n) \\ &\quad - F_1^n(\overline{x_k^{n-1}}; \overline{\theta_r^n}) + O(\epsilon^2) \quad n = 1, 2, \dots, N \end{aligned} \quad (7)$$

Expanding the first difference on the right side of equation

(7) in powers of ϵy_1^n about $\overline{x_k^{n-1}}$ gives:

$$\begin{aligned} F_1^n(x_k^{n-1}; \theta_r^n) - F_1^n(\overline{x_k^{n-1}}; \theta_r^n) &= \sum_{j=1}^S \epsilon y_j^{n-1} \frac{\partial F_1^n(\overline{x_k^{n-1}}; \theta_r^n)}{\partial \overline{x_j^{n-1}}} \\ &\quad + O(\epsilon^2) \end{aligned} \quad (8)$$

Again, $\frac{F_1^n(\overline{x_k^{n-1}}; \theta_r^n)}{\overline{x_j^{n-1}}}$ can be expanded in powers of $\epsilon \varphi^n$ about $\overline{\theta_r^n}$:

$$\frac{\partial F_i^n(\overline{x_k^{n-1}}; \overline{\theta_r^n})}{\partial \overline{x_j^{n-1}}} = \frac{\partial F_i^{n-1}(\overline{x_k^{n-1}}; \overline{\theta_r^n})}{\partial \overline{x_j^{n-1}}} + \sum_{p=1}^t \epsilon_p^n \frac{\partial^2 F_i^n(\overline{x_k^{n-1}}; \overline{\theta_r^n})}{\partial \overline{x_j^{n-1}} \partial \overline{\theta_p^n}} + O(\epsilon^2) \quad (9)$$

Assuming that $F_i^n(\overline{x_k^{n-1}}; \overline{\theta_r^n})$ depend smoothly on $\overline{x_k^{n-1}}$ and $\overline{\theta_r^n}$ so that the terms $O(\epsilon^2)$ may be dropped, then combinations of equations (7), (8), and (9) gives:

$$\epsilon y_i^n = \sum_{j=1}^S \epsilon y_j^{n-1} \frac{\partial F_i^n(\overline{x_k^{n-1}}; \overline{\theta_r^n})}{\partial \overline{x_j^{n-1}}} + F_i^n(\overline{x_k^{n-1}}; \overline{\theta_r^n}) - F_i^n(\overline{x_k^{n-1}}; \overline{\theta_r^n})$$

$$i = 1, 2, \dots, S; \quad n = 1, 2, \dots, N \quad (10)$$

A new set of variables z_i^n is defined to satisfy the difference equations:

$$z_i^{n-1} = \sum_{j=1}^S \frac{\partial F_j^n(\overline{x_k^{n-1}}; \overline{\theta_r^n})}{\partial \overline{x_i^{n-1}}} z_j^n \quad (11)$$

$$i = 1, 2, \dots, S; \quad n = 1, 2, \dots, N$$

Multiplying equation (10) by z_i^n yields:

$$\epsilon y_i^n z_i^n = \sum_{j=1}^S \epsilon y_j^{n-1} \frac{\partial F_i^n(\overline{x_k^{n-1}}; \overline{\theta_r^n})}{\overline{x_j^{n-1}}} z_i^n + z_i^n \left(F_i^n(\overline{x_k^{n-1}}; \overline{\theta_r^n}) - F_i^n(\overline{x_k^{n-1}}; \overline{\theta_r^n}) \right) \quad (12)$$

$$i = 1, 2, \dots, S; \quad n = 1, 2, \dots, N$$

Summing both sides of the above equation gives:

$$\begin{aligned} \sum_{n=1}^N \sum_{i=1}^S \epsilon y_i^n z_i^n &= \sum_{n=1}^N \sum_{i=1}^S \sum_{j=1}^S \epsilon y_j^{n-1} \frac{\partial F_i^n(\overline{x_k^{n-1}}; \overline{\theta_r^n})}{\overline{x_j^{n-1}}} z_i^n \\ &+ \sum_{n=1}^N \sum_{i=1}^S z_i^n \left(F_i^n(\overline{x_k^{n-1}}; \overline{\theta_r^n}) - F_i^n(\overline{x_k^{n-1}}; \overline{\theta_r^n}) \right) \end{aligned} \quad (13)$$

But the first term of the right side of the above equation can be reduced by the following steps:

$$\begin{aligned} \sum_{n=1}^N \sum_{i=1}^S \sum_{j=1}^S \epsilon y_j^{n-1} \frac{\partial F_i^n(\overline{x_k^{n-1}}; \overline{\theta_r^n})}{\overline{x_j^{n-1}}} z_i^n \\ = \sum_{n=1}^N \sum_{j=1}^S \epsilon y_j^{n-1} \sum_{i=1}^S \frac{\partial F_i^n(\overline{x_k^{n-1}}; \overline{\theta_r^n})}{\overline{x_j^{n-1}}} z_i^n \\ = \sum_{n=1}^N \sum_{j=1}^S \epsilon y_j^{n-1} z_j^{n-1} = \sum_{n=1}^N \sum_{i=1}^S \epsilon y_i^{n-1} z_i^{n-1} \end{aligned} \quad (14)$$

Substituting equation (14) into equation (13) and rearranging yields:

$$\epsilon \sum_{n=1}^N \sum_{i=1}^S (y_i^n z_i^n - y_i^{n-1} z_i^{n-1}) = \sum_{n=1}^N \sum_{i=1}^S z_i^n \left[F_i^n(x_k^{n-1}; \theta_r^n) - F_i^n(x_k^{n-1}; \theta_r^n) \right]$$

Hence, the equation may be simplified to:

$$\epsilon \sum_{i=1}^S (y_i^N z_i^N - y_i^0 z_i^0) = \sum_{n=1}^N \sum_{i=1}^S z_i^n \left[F_i^n(x_k^{n-1}; \theta_r^n) - F_i^n(x_k^{n-1}; \theta_r^n) \right] \quad (15)$$

Combining equations (2), (4), and (6) gives:

$$\epsilon y_i^0 = F_i^0(x_k^f; x_k^N) - F_i^0(x_k^f; x_k^N) + O(\epsilon^2) \quad (16)$$

$$i = 1, 2, \dots, S$$

Expanding the difference in powers of ϵy^N about x^N and assuming that $F_i^0(x_k^f, x_k^N)$ depend smoothly on x_k^N so that the terms $O(\epsilon^2)$ can be dropped, it is obtained that:

$$\epsilon y_i^0 = \sum_{j=1}^S \epsilon y_j^N \frac{\partial F_i^0(x_k^f; x_k^N)}{\partial x_j^N} \quad (17)$$

Substituting equation (17) into (15) gives:

$$\epsilon \left(\sum_{i=1}^S y_i^N z_i^N - \sum_{i=1}^S \sum_{j=1}^S z_i^0 y_j^N \frac{\partial F_i^0(x_k^f; \overline{x_k^N})}{\partial \overline{x_j^N}} \right) = \sum_{n=1}^N \sum_{i=1}^S z_i^n$$

$$\left(F_i^n(x_k^{\overline{n-1}}; \theta_r^n) - F_i^n(x_k^{\overline{n-1}}; \overline{\theta_r^n}) \right)$$

Interchanging the dummy variables, i and j , in the second term on the left side yields:

$$\epsilon \sum_{i=1}^S y_i^N \left(z_i^N - \sum_{j=1}^S z_j^0 \frac{\partial F_j^0(x_k^f; \overline{x_k^N})}{\partial \overline{x_i^N}} \right) = \sum_{n=1}^N \sum_{i=1}^S z_i^n \left(F_i^n(x_k^{\overline{n-1}}; \theta_r^n) - F_i^n(x_k^{\overline{n-1}}; \overline{\theta_r^n}) \right) \quad (18)$$

By imposing on the z 's the conditions

$$z_i^N - \sum_{j=1}^S z_j^0 \frac{\partial F_j^0(x_k^f; \overline{x_k^N})}{\partial \overline{x_i^N}} = \delta_{im} \quad (19)$$

where δ_{im} is the Kronecker delta, equation (18) becomes:

$$\epsilon y_m^N = \sum_{n=1}^N \sum_{i=1}^S z_i^n \left(F_i^n(x_k^{\overline{n-1}}; \theta_r^n) - F_i^n(x_k^{\overline{n-1}}; \overline{\theta_r^n}) \right) \quad (20)$$

The $\overline{\theta_r^n}$ being the sequence that minimizes x_m^N , the effect of the perturbation represented by equation (5) can only be to make

$$\epsilon y_m^N \geq 0 \quad (21)$$

Combining equations (20) and (21) gives:

$$\sum_{n=1}^N \sum_{i=1}^S z_i^n \left(F_i^n(x_k^{n-1}; \theta_r^n) - F_i^n(x_k^{n-1}; \bar{\theta}_r^n) \right) \geq 0 \quad (22)$$

$$n = 1, 2, \dots, N$$

Now in equation (22) the perturbed control actions θ_r^n are independent of each other. It may be concluded that each term of equation (22) containing a set of independent variables θ_r^n must itself be non-negative; thus

$$\sum_{i=1}^S z_i^n \left(F_i^n(x_k^{n-1}; \theta_r^n) - F_i^n(x_k^{n-1}; \bar{\theta}_r^n) \right) \geq 0 \quad (23)$$

Hence the procedure for finding the minimizing sequence θ_p^n , $p = 1, 2, \dots, t$, is to choose θ_p^n to minimize the function

$$\sum_{i=1}^S z_i^n F_i^n(x_k^{n-1}; \theta_r^n)$$

That is, θ_r^n must be selected from either (1) those satisfying

$$\sum_{i=1}^S z_i^n \frac{\partial F_i^n(x_k^{n-1}; \theta_r^n)}{\partial \theta^n} = 0, \text{ when } \zeta_p^n < \theta_p^n < \gamma_p^n$$

$$\text{or (2) } \theta_p^n = \zeta_p^n$$

$$\text{or (3) } \theta_p^n = \gamma_p^n$$

whichever makes $\sum_{i=1}^S z_i^n F_i^n(x_k^{n-1}; \theta_r^n)$, $n = 1, 2, \dots, N$ and

$p = 1, 2, \dots, t$, the smallest.

The general algorithm for finding the minimizing sequence of control actions can be written down compactly in terms of the Hamiltonian formalism introduced by Pontryagin (13).

Let

$$H^n = \sum_{j=1}^S z_j^n F_j^n(x_k^{n-1}; \theta_r^n)$$

then:

$$x_i^n = \frac{\partial H^n}{\partial z_i^n}, \quad x_i^0 = F_i^0(x_i^f; x_i^n)$$

$$z_i^{n-1} = \frac{\partial H^n}{\partial x_i^{n-1}}, \quad z_i^N - \sum_{j=1}^S z_j^0 \frac{\partial F_j^0(x_k^f; x_k^N)}{\partial x_i^N} = \delta_{im}$$

θ_p^n are determined where $H^n = \min$

$n = 1, 2, \dots, N; \quad i = 1, 2, \dots, S;$

$p = 1, 2, \dots, t$

(24)

If the maximizing sequence of control actions instead of the minimizing sequence is to be decided, the basic algorithm, equation (24), remains unchanged except that the equation $H^n = \min$ is replaced by $H^n = \max$.

It can be seen that the simplification in computation introduced by the maximum principle is similar to that by dynamic programming. Instead of choosing all N sets of θ_r 's simultaneously, only one set is determined at a time.

With some modification, this basic algorithm is applicable to a variety of problems. The modifications required in two typical cases will be briefly outlined.

1. Problems with fixed end point specifications.

Suppose that, for the system described by equation (1) and (2), it is desired to choose the sequence of the sets of control actions θ_p^n , $n = 1, 2, \dots, N$; $p = 1, 2, \dots, t$, to minimize x_m^N , while keeping the final values of certain others of the x 's at preassigned values, for example

$$x_3^N = a_3^N, \quad x_7^N = a_7^N \quad (25)$$

To solve this, the conditions, equation (19), are replaced by:

$$z_i^N - \sum_{j=1}^S z_j^0 \frac{\partial F_j^0(x_k^f, x_k^N)}{\partial x_i^N} = \begin{cases} 1, & i = m \\ 0, & i \neq m, 3, 7 \end{cases} \quad (26)$$

The missing conditions for $i = 3, 7$ are made up by equation (25). This modification is verified by noting that

$$y_3^N = y_7^N = 0$$

and thus the conditions $z_i^N - \sum_{j=1}^S z_j^0 \frac{\partial F_j^0(x_k^f, x_k^N)}{\partial x_i^N} = 0$ for $i = 3, 7$

included in equation (19) are redundant for the isolation of y_m^N from equation (18).

2. Problems with choice of initial values.

Suppose that some of the initial values of the x 's, say x_1^f , x_2^f , and x_5^f are not prescribed, and it is desired to choose these missing initial values as well as the sequence of control actions, to minimize x_m^N . The problem may be solved by imposing an additional condition,

$$x_j^f \sum_{i=1}^5 z_i^0 \frac{\partial F_i^0(x_k^f; x_k^N)}{\partial x_j^f} = \min, \quad j = 1, 2, 5 \quad (27)$$

to the basic algorithm, equation (24).

The addition of this condition is confirmed by observing that when x_i^f , $i = 1, 2, 5$, are not preassigned, equations (4), (16), and (17) must be modified to:

$$\overline{x_i^0} = F_i^0(\overline{x_k^f}, \overline{x_k^N}) \quad (28)$$

$$\epsilon y_i^0 = F_i^0(x_k^f, x_k^N) - F_i^0(\overline{x_k^f}, \overline{x_k^N}) + O(\epsilon^2) \quad (29)$$

and

$$\epsilon y_i^0 = \sum_{j=1}^5 \epsilon y_j^N \frac{\partial F_i^0(\overline{x_k^f}, \overline{x_k^N})}{\partial x_j^N} + \sum_{j=1, 2, 5} \epsilon y_j^f \frac{\partial F_i^0(\overline{x_k^f}, \overline{x_k^N})}{\partial x_j^f} \quad (30)$$

$$\text{where } \epsilon y_j^f = x_j^f - \overline{x_j^f}$$

Accordingly, equation (23) becomes:

$$\sum_{i=1}^S z_i^n \left(F_i^n (\overline{x_k^{n-1}}; \theta_r^n) - F_i^n (\overline{x_k^{n-1}}; \overline{\theta_r^n}) \right) + \epsilon \sum_{j=1, 2, 5} y_j^f.$$

$$\sum_{i=1}^S z_i^0 \frac{\partial F_i^0 (\overline{x_k^f}; \overline{x_k^N})}{\partial \overline{x_j^f}} \geq 0$$

Since y_j^f , $j = 1, 2, 5$ are independent of each other and of θ_p^n , it may be concluded that

$$\epsilon y_j^f \sum_{i=1}^S z_i^0 \frac{\partial F_i^0 (\overline{x_k^f}; \overline{x_k^N})}{\partial \overline{x_j^f}}, \quad j = 1, 2, 5 \text{ must all be non-}$$

negative. That is

$$x_j^f \sum_{i=1}^S z_i^0 \frac{\partial F_i^0 (\overline{x_k^f}; \overline{x_k^N})}{\partial \overline{x_j^f}} \leq x_j^f \sum_{i=1}^S z_i^0 \frac{\partial F_i^0 (\overline{x_k^f}; \overline{x_k^N})}{\partial \overline{x_j^f}}$$

$$j = 1, 2, 5$$

which are equivalent to equation (27).

For the process without product recycle, r is equal to zero, and equation (2) reduces to

$$x_i^0 = F_i^0 (x_k^f; x_k^N) = x_i^f \quad i = 1, 2, \dots, S \quad (2a)$$

and the basic algorithm, equation (24), becomes:

$$H^n = \sum_{j=1}^S z_j^n F_j^n(x_k^{n-1}; \theta_r^n)$$

$$x_i^n = \frac{\partial H^n}{\partial z_i^n}, \quad x_i^0 = x_i^f$$

$$z_i^{n-1} = \frac{\partial H^n}{\partial x_i^{n-1}}, \quad z_i^N = \delta_{im}$$

$$\theta_p^n \text{ are determined where } H^n = \min$$

$$n = 1, 2, \dots, N; \quad i = 1, 2, \dots, S;$$

$$p = 1, 2, \dots, t$$

(24a)

The modified conditions, equation (26), for the problem with fixed end point specifications become:

$$z_i^N = \begin{cases} 1, & i = m \\ 0, & i \neq m, 3, 7 \end{cases} \quad (26a)$$

The additional condition, equation (27), required by the problem with choices of initial values, is changed to:

$$x_j^f z_j^0 = \min \quad j = 1, 2, 5 \quad (27a)$$

APPLICATION TO ONE-DIMENSIONAL SYSTEMS

In this section the general working equations will be derived, and the general computational scheme will be discussed for a special class of one-dimensional systems. Several concrete examples will be worked out in detail to illustrate the use of the algorithm.

General Equations: The special class of one-dimensional systems which are to be investigated is a homogeneous system with product recycle, containing one state variable described by

$$x_1^n = F(x_1^{n-1}; \theta^n) \quad n = 1, 2, \dots, N \quad (31)$$

in which a sum of a certain arbitrary function of x_1 , over all stages of the system, such as $\sum_{n=1}^N G(x_1^{n-1}; \theta^n)$, is to be maximized. The mixing condition is given by

$$x_1^0 = \frac{q x_1^f + r x_1^N}{q + r} \quad (32)$$

This problem can be reduced to the standard form by introducing a new state variable, x_2 , satisfying

$$x_2^n = x_2^{n-1} + G(x_1^{n-1}; \theta^n), \quad x_2^0 = 0 \quad (33)$$

$$n = 1, 2, \dots, N$$

It can be shown that $\sum_{n=1}^N G(x_1^{n-1}; \theta^n) = x_2^N$. Now the problem is transformed into one in which it is desired to maximize x_2^N of a system described by the performance equation represented by equations (31) and (33).

According to the basic algorithm, equation (24), a new set of variables is introduced:

$$\left. \begin{aligned} z_1^{n-1} &= \frac{\partial F(x_1^{n-1}; \theta^n)}{\partial x_1^{n-1}} z_1^n + \frac{\partial G(x_1^{n-1}; \theta^n)}{\partial x_1^{n-1}} z_2^n \\ z_2^{n-1} &= z_2^n; \quad n = 1, 2, \dots, N \end{aligned} \right\} \quad (34)$$

with the final conditions being:

$$z_1^N = \frac{r}{q+r} z_1^0 \quad z_2^N = 1 \quad (35)$$

Substituting equation (35) into equation (34) gives:

$$z_2^n = 1; \quad n = 1, 2, \dots, N \quad (36)$$

and

$$\left. \begin{aligned} z_1^{n-1} &= \frac{\partial F(x_1^{n-1}; \theta^n)}{\partial x_1^{n-1}} z_1^n + \frac{\partial G(x_1^{n-1}; \theta^n)}{\partial x_1^{n-1}} \\ n &= 1, 2, \dots, N \end{aligned} \right\} \quad (37)$$

The Hamiltonian becomes:

$$H^n = z_1^n F(x_1^{n-1}; \theta^n) + G(x_1^{n-1}; \theta^n) + x_2^{n-1}$$

Assuming that the stationary solutions will represent the maxima, θ^n may be found where:

$$\frac{\partial H^n}{\partial \theta^n} = z_1^n \frac{\partial F(x_1^{n-1}; \theta^n)}{\partial \theta^n} + \frac{\partial G(x_1^{n-1}; \theta^n)}{\partial \theta^n} = 0$$

The solution for z_1^n is:

$$z_1^n = - \frac{\frac{\partial G(x_1^{n-1}; \theta^n)}{\partial \theta^n}}{\frac{\partial F(x_1^{n-1}; \theta^n)}{\partial \theta^n}} \quad n = 1, 2, \dots, N \quad (38)$$

Insertion of equation (38) into equation (37) gives:

$$\frac{\frac{\partial G(x_1^{n-1}; \theta^n)}{\partial \theta^n}}{\frac{\partial F(x_1^{n-1}; \theta^n)}{\partial \theta^n}} = \frac{\frac{\partial G(x_1^n; \theta^{n+1})}{\partial \theta^{n+1}}}{\frac{\partial F(x_1^n; \theta^{n+1})}{\partial \theta^{n+1}}} \cdot \frac{\frac{\partial F(x_1^n; \theta^{n+1})}{\partial x_1^n}}{\frac{\partial G(x_1^n; \theta^{n+1})}{\partial x_1^n}} \quad n = 1, 2, \dots, N-1 \quad (39)$$

and

$$z_1^0 = - \frac{\frac{\partial G(x_1^0; \theta^1)}{\partial \theta^1}}{\frac{\partial F(x_1^0; \theta^1)}{\partial \theta^1}} \cdot \frac{\frac{\partial F(x_1^0; \theta^1)}{\partial x_1^0}}{\frac{\partial G(x_1^0; \theta^1)}{\partial x_1^0}} \quad (39a)$$

Equation (39) can be written in a compact form in terms of Jacobian as follows:

$$\frac{\frac{\partial G(x_1^{n-1}; \theta^n)}{\partial \theta^n}}{\frac{\partial F(x_1^{n-1}; \theta^n)}{\partial \theta^n}} = \frac{\frac{\partial (G(x_1^n; \theta^{n+1}), F(x_1^n; \theta^{n+1}))}{\partial (\theta^{n+1}, x_1^n)}}{\frac{\partial F(x_1^n; \theta^{n+1})}{\partial \theta^{n+1}}}$$

Substituting equations (38), (39), and (39a) into equation (35) gives:

$$-\frac{\frac{\partial G(x_1^{N-1}; \theta^N)}{\partial \theta^N}}{\frac{\partial F(x_1^{N-1}; \theta^N)}{\partial \theta^N}} = \frac{r}{q+r} \left[\frac{\partial G(x_1^0; \theta^1)}{\partial x_1^0} - \frac{\partial F(x_1^0; \theta^1)}{\partial x_1^0} \cdot \frac{\frac{\partial G(x_1^0; \theta^1)}{\partial \theta^1}}{\frac{\partial F(x_1^0; \theta^1)}{\partial \theta^1}} \right] \quad (40)$$

By assigning values to both x_1^{N-1} and θ^N , the value for x_1^N can be computed from equation (31). The result is then substituted into equation (32) to calculate x_1^0 , which is, in turn, inserted into equation (40) to obtain θ^1 . The values for both x_1^0 and θ^1 can also be calculated by iterative applications of equations (31) and (39). The procedure is repeated until the values of x_1^0 and θ^1 computed by both ways are equal.

If the process is operated without recycling, r vanishes and equation (40) reduces to

$$\frac{\partial G(x_1^{N-1}; \theta^N)}{\partial \theta^N} = 0 \quad (40a)$$

Then, by assigning a value to x_1^N , both x_1^{N-1} and θ^N can be

obtained by solving equations (31) and (40a) simultaneously; and the corresponding values for x_1^0 can be calculated by iterative uses of equations (31) and (39). The procedure is repeated until the computed value of x_1^0 is equal to the initial value x_1^f . It is worthwhile to note that for each assigned value of x_1^N , the corresponding values of θ_1^n , $n = 1, 2, \dots, N$ computed are the optimal control actions corresponding to the initial condition x_1^f obtained in each run of trial calculations.

For the problem with a prescribed end point x_1^N , equation (40) is deleted for both recycling and non-recycling processes. With x_1^N given, the value for θ_1^N can be computed from equation (31) by assigning a value to x_1^{N-1} . The corresponding value for x_1^0 is obtained by iterative utilizations of equations (31) and (39). The result is directly checked with the given x_1^f for the non-recycling process. For the recycling process, the computed x_1^0 is substituted into equation (32) to calculate x_1^f , which is then compared with the given value. The trial calculations are repeated until the computed value matches the given one. As in the case of a free end point, the sequence of θ_1^n , $n = 1, 2, \dots, N$ computed, for each assumed value of x_1^{N-1} , is the optimal sequence corresponding to the initial condition x_1^f obtained in each run of trial calculations.

The application of these working equations and the computational scheme will be illustrated in the following two examples.

Cross-current Extraction Process: The problem of wash

water allocation in a stagewise cross-current extraction system without recycle has been solved with dynamic programming by Amundson, Aris, and Rudd (3), and with a non-imbedding technique by Converse (14). As has been stated before, the method of dynamic programming has the disadvantages that it requires a large amount of computer memory and that it may introduce a considerable error due to interpolation. While Converse's non-imbedding technique may reduce the memory requirements and avoid interpolation, its computation time is much longer than that of dynamic programming and will increase rapidly with the increase in the number of stages in the system.

Rudd and Blum (15) have shown that the dynamic programming tables for the process without recycle obtained by Aris et al. (3) can be used without modification to determine the optimal operating conditions for a process with recycle. It was recently pointed out by Jackson (16) that Rudd's method is in fact fallacious. Jackson discredited the assumption, made by Rudd and Blum, that the process of determining optimal conditions in the recycle system is mathematically equivalent to the one of optimizing the sub-process of the forward stream and matching conditions at the recycle point. However, it should be noted that the first part of this assumption is not to be doubted. The incorrect results arise from assigning values to the initial conditions for optimizing the sub-process. The assignment of specific values to the initial conditions in each trial calculation is tantamount to assuming that the initial conditions are independent of the choice of operating conditions, which is

contrary to the fact. Actually, optimizing a recycle process is mathematically equivalent to optimizing the forward sub-process with variable initial conditions, which are dependent on the operating conditions.

In the following passages, the optimization problem of the process with product recycle will be solved by the application of the recursion relation of equation (39), and it will be shown that the process without recycle is a special case of the process with recycle.

A schematic diagram of the process under consideration is presented in Fig. 2. The process consists of N equilibrium stages through which a solvent containing a solute passes. A portion of the end product is fed back to the first stage, at a flow rate r . The solute is extracted from the solvent by the addition of wash water at each stage. The solvent and wash water are assumed immiscible. The solvent flows from stage to stage at a rate $(q + r)$. The state variable is the concentration of solute x_1 . The control variable at each stage is the amount of wash water w . The performance equation may be obtained by a material balance for the solute about the n^{th} stage:

$$x_1^n = x_1^{n-1} - v^n u^n; \quad n = 1, 2, \dots, N \quad (41)$$

Here $v^n = w^n/(q + r)$, and u^n is the concentration of solute in the wash water leaving the n^{th} stage, which is in equilibrium with the raffinate.

The gross return from the process is the total amount of solute extracted by the wash water. The costs are those

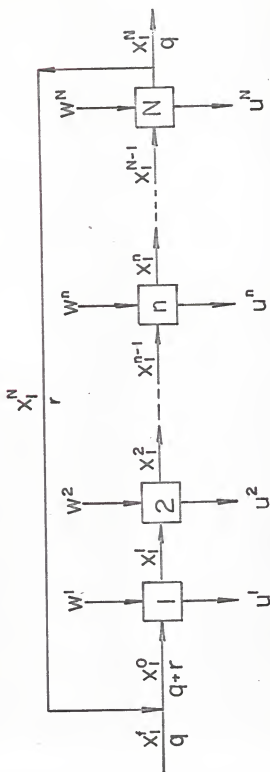


Fig.2. Cross-current extraction process with recycle.

associated with the wash water. The net profit, which is to be maximized, is then:

$$P = q (x_1^f - x_1^N) - \lambda \sum_{n=1}^N w^n \quad (42)$$

where λ is the relative cost of wash water.

The solute concentration of the mixed stream which enters the first stage of the process is:

$$x_1^0 = \frac{q x_1^f + r x_1^N}{q + r} \quad (43)$$

Solving equation (43) for x_1^f , and substituting the result into equation (42) gives:

$$P = (q + r) (x_1^0 - x_1^N) - \lambda \sum_{n=1}^N w^n \quad (44)$$

which is the net profit for the sub-process within the recycle stream. It is clear that the optimization problem for the process with recycle is equivalent to the problem for the sub-process within the recycle stream, in which it is desired to maximize P in equation (44), by the proper choices of w^n , $n = 1, 2, \dots, N$, subject to the initial condition, equation (43), and the performance equation, represented by equation (41).

From equation (41) it is seen that

$$\sum_{n=1}^N (x_1^{n-1} - x_1^n) = x_1^0 - x_1^N = \sum_{n=1}^N v^n u^n$$

Inserting this into equation (44) gives:

$$\frac{P}{q+r} = \sum_{n=1}^N v^n (u^n - \lambda) \quad (45)$$

By defining a new set of state variables x_2^n satisfying

$$x_2^n = x_2^{n-1} + v^n (u^n - \lambda); \quad x_2^0 = 0 \quad (46)$$

$$n = 1, 2, \dots, N$$

it can be shown that $\frac{P}{q+r} = x_2^N$. Thus the optimization problem is now transformed into one in which x_2^N is to be maximized by the proper choices of v^n , $n = 1, 2, \dots, N$, in a system described by equations (41), (43), and (46).

If the phase equilibrium relation is given by

$$u = \bar{\Phi}(x_1) \quad (47)$$

equations (41) and (46) can be rewritten as:

$$x_1^n = F(x_1^{n-1}; v^n) \quad (48)$$

$$x_2^n = x_2^{n-1} + x_1^{n-1} - F(x_1^{n-1}; v^n) - \lambda v^n \quad (49)$$

$$n = 1, 2, \dots, N$$

Comparing equation (49) with equation (33), it is seen that

$$G(x_1^{n-1}; v^n) = x_1^{n-1} - F(x_1^{n-1}; v^n) - \lambda v^n \quad (50)$$

Differentiating equation (41) with respect to v^n gives:

$$\frac{\partial x_1^n}{\partial v^n} = -u^n - v^n \frac{\partial u^n}{\partial x_1^n} \frac{\partial x_1^n}{\partial v^n} \quad (51)$$

Substituting equations (47) and (48) into (51), and then solving

for $\frac{\partial F(x_1^{n-1}; v^n)}{\partial v^n}$ gives:

$$\frac{\partial F(x_1^{n-1}; v^n)}{\partial v^n} = - \frac{\Phi(x_1^n)}{1 + v^n \frac{d\Phi(x_1^n)}{dx_1^n}} \quad (52)$$

The expression for $\frac{\partial F(x_1^{n-1}; v^n)}{\partial x_1^{n-1}}$ may be obtained in a similar

way. Thus, differentiating equation (41) with respect to x_1^{n-1} and then combining with equations (47) and (48) gives:

$$\frac{\partial F(x_1^{n-1}; v^n)}{\partial x_1^{n-1}} = \frac{1}{1 + v^n \frac{d\Phi(x_1^n)}{dx_1^n}} \quad (53)$$

The expressions for $\frac{\partial G(x_1^{n-1}; v^n)}{\partial v^n}$ and $\frac{\partial G(x_1^{n-1}; v^n)}{\partial x_1^{n-1}}$ are obtained

by differentiating equation (50) with respect to v^n and x_1^{n-1} and

then substituting for $\frac{\partial F(x_1^{n-1}; v^n)}{\partial v^n}$ and $\frac{\partial G(x_1^{n-1}; v^n)}{\partial x_1^{n-1}}$ from

equations (52) and (53) respectively. The results are:

$$\begin{aligned} \frac{\partial G(x_1^{n-1}; v^n)}{\partial v^n} &= - \frac{\partial F(x_1^{n-1}; v^n)}{\partial v^n} - \lambda \\ &= \frac{\Phi(x_1^n)}{1 + v^n \frac{d\Phi(x_1^n)}{dx_1^n}} - \lambda \end{aligned} \quad (54)$$

$$\begin{aligned} \frac{\partial G(x_1^{n-1}; v^n)}{\partial x_1^{n-1}} &= 1 - \frac{\partial F(x_1^{n-1}; v^n)}{\partial x_1^{n-1}} \\ &= 1 - \frac{1}{1 + v^n \frac{d\Phi(x_1^n)}{dx_1^n}} \end{aligned} \quad (55)$$

Inserting equations (52) through (55) into equation (39) yields:

$$1 + v^n \frac{d\Phi(x_1^n)}{dx_1^n} = \frac{\Phi(x_1^n)}{\Phi(x_1^{n+1})} \quad (56)$$

Solving equation (41) for v^n gives:

$$v^n = \frac{x_1^{n-1} - x_1^n}{\Phi(x_1^n)} \quad (57)$$

Substituting equation (57) into equation (56) and solving for x_1^{n-1} gives the following recursion relation:

$$x_1^{n-1} = x_1^n + \frac{\bar{\Phi}(x_1^n)}{\frac{d\bar{\Phi}(x_1^n)}{dx_1^n}} \left[\frac{\bar{\Phi}(x_1^n)}{\bar{\Phi}(x_1^{n+1})} - 1 \right] \quad (58)$$

$$n = 1, 2, \dots, N-1$$

Inserting equations (52) through (55) into equation (40), and then combining with equation (57) yields:

$$x_1^{N-1} = x_1^N + \frac{[\bar{\Phi}(x_1^N)]^2}{\frac{d\bar{\Phi}(x_1^N)}{dx_1^N}} \left\{ \left[\frac{1}{\lambda} - \frac{1}{\bar{\Phi}(x_1^N)} \right] - \frac{r}{q+r} \left[\frac{1}{\lambda} - \frac{1}{\bar{\Phi}(x_1^1)} \right] \right\} \quad (59)$$

The computational procedure for finding the optimal sequence of v^n , $n = 1, 2, \dots, N$, is similar to that given in the paragraphs following equation (40). However, the calculation for this particular case is simpler, because equation (58) contains only one state variable. For a 3-stage process, equation (59) becomes:

$$x_1^2 = x_1^3 + \frac{[\bar{\Phi}(x_1^3)]^2}{\frac{d\bar{\Phi}(x_1^3)}{dx_1^3}} \left\{ \left[\frac{1}{\lambda} - \frac{1}{\bar{\Phi}(x_1^3)} \right] - \frac{r}{q+r} \left[\frac{1}{\lambda} - \frac{1}{\bar{\Phi}(x_1^1)} \right] \right\} \quad (59a)$$

Letting $n = 1$ in equation (58) and inserting for x_1^0 from equation (43) gives:

$$\frac{q x_1^f + r x_1^3}{q + r} = x_1^1 + \frac{\frac{\Phi(x_1^1)}{d\Phi(x_1^1)}}{\frac{dx_1^1}{dx_1^1}} \left[\frac{\Phi(x_1^1)}{\Phi(x_1^2)} - 1 \right] \quad (60)$$

When $n = 2$, equation (58) gives:

$$x_1^1 = x_1^2 + \frac{\frac{\Phi(x_1^2)}{d\Phi(x_1^2)}}{\frac{2}{dx_1^1}} \left[\frac{\Phi(x_1^2)}{\Phi(x_1^3)} - 1 \right] \quad (61)$$

Now it is seen that these three equations contain only three independent variables, namely, x_1^1 , x_1^2 , and x_1^3 . Hence, the simultaneous solutions of these equations will give the optimal values for state variables x_1^1 , x_1^2 , and x_1^3 . The value for x_1^0 can be recovered from equation (43). The optimal wash water allocation is then determined by equation (57).

It is clear that the process without recycle may be reduced from the process with recycle by letting $r = 0$. Thus the initial condition, equation (43), becomes

$$x_1^0 = x_1^f$$

and equation (59) reduces to

$$x_1^{N-1} = x_1^N + \frac{\frac{[\Phi(x_1^N)]^2}{d\Phi(x_1^N)}}{\frac{dx_1^N}{dx_1^N}} \left[\frac{1}{\lambda} - \frac{1}{\Phi(x_1^N)} \right] \quad (59b)$$

while the recursion relation, equation (58), remains unaltered. The computational procedure is simplified considerably for non-recycling process. By assigning a value to x_1^N , the value for x_1^{N-1} is computed by equation (59b) and the corresponding value for x_1^0 is obtained by the iterative uses of equation (58). The trial calculations are repeated until the computed value of x_1^0 becomes equal to x_1^f . As before, the values of x_1^n , $n = 1, 2, \dots, N$, computed in each run of the trial calculations are the optimal state variables corresponding to the initial conditions, x_1^0 , obtained in each run. The optimal values of v^n are then recovered from equation (57).

The numerical examples illustrated in References (3) and (15) are recalculated here by means of equations (58) through (61). The phase equilibrium data used are given in Table 1. In Table 2 the results are compared with those of References (3) and (15). For the process without recycle, the slight discrepancy between the results obtained by the two different methods may be due to the fact that slightly different equilibrium data are used in the calculations and may also be due to interpolation error inherent in the dynamic programming. It has been mentioned that the results obtained by Rudd and Blum for the process with recycle were incorrect, and it can be seen from the table that the profit calculated by the maximum principle is 3.4 percent higher than that computed by Rudd and Blum.

The simple case with a linear equilibrium relation $x_1 = u$ can be solved analytically. Equation (47) becomes:

$$u = \bar{p}(x_1) = \frac{x_1}{\mu}$$

TABLE 1
Equilibrium Data

<u>u</u>	<u>x</u>
0.000	0.000
0.027	0.010
0.050	0.020
0.073	0.030
0.094	0.040
0.119	0.050
0.138	0.060
0.153	0.070
0.163	0.080
0.170	0.090
0.173	0.100
0.176	0.110
0.178	0.120
0.179	0.130
0.179	0.140
0.180	0.150
0.180	0.160
0.182	0.170
0.186	0.180
0.192	0.190
0.200	0.200

TABLE 2

Comparison of the Results

Obtained by the Maximum Principle and Dynamic Programming

	<u>Maximum Principle</u>	<u>Dynamic Programming</u>
<u>(1) Without recycle</u>		
<u>$a = 1, x_1^f = 0.20, \lambda = 0.05$</u>		
<u>Wash Water Allocation</u>		
Stage 1, w_1	0.657	0.647
Stage 2, w_2	0.279	0.284
Stage 3, w_3	0.263	0.260
<u>Profit</u>	0.1071	0.1075
<u>(2) With recycle</u>		
<u>$a = 1, r = 1, x_1^f = 0.20, \lambda = 0.05$</u>		
<u>Wash Water Allocation</u>		
Stage 1, w_1	0.5105	0.610
Stage 2, w_2	0.3339	0.440
Stage 3, w_3	0.2977	0.420
<u>Profit</u>	0.1007	0.0974

Substituting this relation into equation (58) gives:

$$x_1^{n-1} = x_1^n + \frac{\frac{x_1^n}{\mu}}{\frac{1}{\mu}} \left[\frac{\frac{-1}{\mu}}{\frac{x_1^{n+1}}{\mu}} - 1 \right]$$

which can be simplified to

$$(x_1^n)^2 = x_1^{n-1} \cdot x_1^{n+1}; \quad n = 1, 2, \dots, N-1$$

The general solution of this finite difference equation is:

$$x_1^n = B(A)^n; \quad n = 0, 1, \dots, N \quad (62)$$

where A and B are two constants to be determined by the initial and final conditions. Thus, substituting equation (62) into equations (43) and (59) gives:

$$E = \frac{q}{q+r} x_1^f + \frac{r}{q+r} B(A)^N \quad (63)$$

and

$$E(A)^{N-1} = B(A)^N + \frac{(B(A)^N)'}{\mu} \left\{ \left[\frac{1}{\lambda} - \frac{\mu}{B(A)^N} \right] - \frac{r}{q+r} \left[\frac{1}{\lambda} - \frac{\mu}{BA} \right] \right\} \quad (64)$$

Combining these two equations yields:

$$\left(\frac{r}{q+r} \right)^2 (A)^{2N} - \left(\frac{q}{q+r} \right)^2 \frac{x_1^f}{\mu \lambda} (A)^{N+1} - 2 \left(\frac{r}{q+r} \right) (A)^N + 1 = 0 \quad (65)$$

and

$$B = \frac{\frac{q}{q+r} \frac{1}{1 - (A)^N}}{1 - \frac{r}{q+r} (A)^N} \quad (66)$$

The value of A may be obtained from equation (65). The result is then substituted into equation (66) to compute B. The v^n are then recovered from equation (57) as

$$v^n = \mu \left(\frac{1}{\lambda} - 1 \right) \quad (67)$$

Since v^n is independent of n, it may be concluded that the optimal policy is to distribute the wash water equally to all stages.

The profit is obtained by inserting equations (62) and (67) into equation (45):

$$\frac{p}{q+r} = \sum_{n=1}^N \mu \left(\frac{1}{\lambda} - 1 \right) \left[\frac{B(A)^n}{\mu} - \lambda \right] = \frac{q}{q+r} \frac{1}{1 - (A)^N} \left[1 - (A)^N \right] - \frac{N\mu\lambda}{A} + N\mu\lambda \quad (68)$$

For the process without recycle, r vanishes and equation (65) reduces to

$$(A)^{N+1} = \frac{\mu\lambda}{x_1}$$

which when solved for A gives:

$$A = \left(\frac{\mu\lambda}{x_1} \right)^{\frac{1}{N+1}} \quad (69)$$

Equation (66) reduces to

$$B = x_1^f$$

Equations (62), (67), and (68) become:

$$x_1^n = x_1^f \left(\frac{\mu \lambda}{x_1^f} \right)^{\frac{n}{N+1}} \quad (70)$$

$$v^n = \left[\frac{(\mu)^N}{\lambda} x_1^f \right]^{\frac{1}{N+1}} - \mu \quad (71)$$

and

$$\frac{p}{q} = x_1^f + N\mu\lambda - (N+1) \left[(\mu)^N (\lambda)^N x_1^f \right]^{\frac{1}{N+1}} \quad (72)$$

The Continuous Flow Stirred Tank Reactor with a Single Reaction: The problem is to choose the sequence of temperature and holding time so that a given conversion be achieved from a given feed state with the least total holding time.

The equation for a single reaction may be written as

$$\sum_{i=1}^S \alpha_i A_i = 0 \quad (73)$$

Here α_i , the stoichiometric coefficient of A_i , is positive if A_i is a product.

If c_i^n is the concentration of A_i in the n^{th} reactor and e^n

is the holding time, a material balance for A_1 about the n^{th} reactor gives (see Fig. 3):

$$c_1^n = c_1^{n-1} + \theta^n \frac{dc_1^n}{dt}; \quad n = 1, 2, \dots, N \quad (74)$$

The extent of reaction x_1 may be defined as:

$$x_1 = \frac{c_1 - c_{10}}{\alpha_1} \quad (75)$$

where c_{10} is the concentration in a certain fixed reference state. Equation (74) can be written in terms of the extent of reaction as:

$$x_1^n = x_1^{n-1} + \theta^n \frac{dx_1^n}{dt}; \quad n = 1, 2, \dots, N \quad (76)$$

Assuming that the pressure is constant, the rate of reaction will be a function of temperature, T , and concentrations, c_i . But from equation (75), it is seen that concentrations are linear functions of the extent of reaction; hence, the rate of reaction can be expressed as a function of temperature and the extent of reaction. Consequently, equation (76) becomes:

$$x_1^n = x_1^{n-1} + \theta^n r(x_1^n; T^n); \quad n = 1, 2, \dots, N \quad (77)$$

By defining

$$x_2^n = x_2^{n-1} + \theta^n; \quad n = 1, 2, \dots, N \quad (78)$$

$$x_2^0 = 0 \quad (79)$$



Fig. 3. The stirred tank reactor sequence.

It can be shown that x_2^N is equal to the total holding time, i.e., $\sum \theta^n$. The optimization problem is now formulated as one in which x_2^N is to be minimized by the proper choices of θ^n and T^n , $n = 1, 2, \dots, N$, with the initial and final conditions:

$$x_1^0 = a_1^0, x_2^0 = 0, \text{ and } x_1^N = a_1^N \quad (80)$$

and the performance equations represented by equations (77) and (78).

It has been known from experience that the optimal temperature policy with a single reaction has a disjoint characteristic, that is, the temperature should always be chosen so that the rate of reaction will be as large as possible at each stage. This can be shown mathematically as follows.

Solving equation (77) for x_1^n gives:

$$x_1^n = F_1(x_1^{n-1}; \theta^n; T^n) \quad (81)$$

The Hamiltonian for the system is:

$$H^n = z_1^n F_1(x_1^{n-1}; \theta^n; T^n) + z_2^n (x_2^{n-1} + \theta^n)$$

The optimal choice of the temperature will be found where

$$\frac{\partial H^n}{\partial T^n} = z_1^n \frac{\partial F_1(x_1^{n-1}; \theta^n; T^n)}{\partial T^n} = 0 \quad (82)$$

Differentiating equation (77) with respect to T^n yields:

$$\frac{\partial F_1(x_1^{n-1}; \theta^n; T^n)}{\partial T^n} = \theta^n \left[\frac{\partial r(x_1^n; T^n)}{\partial T^n} + \frac{\partial r(x_1^n; T^n)}{\partial x_1^n} \cdot \frac{\partial F_1(x_1^{n-1}; \theta^n; T^n)}{\partial T^n} \right] \quad (83)$$

By combining equations (82) and (83) and noting that $\theta^n \neq 0$ and $x_1^n \neq 0$, it may be concluded that

$$\frac{\partial r(x_1^n; T^n)}{\partial T^n} = 0 \quad (84)$$

This shows that the temperature should be chosen so that the rate of reaction be as large as possible at each stage. Denoting this maximum value by $R(x_1^n)$, equations (77) and (81) become:

$$x_1^{n-1} = x_1^n - \theta^n R(x_1^n); \quad n = 1, 2, \dots, N \quad (85)$$

$$x_1^n = F(x_1^{n-1}; \theta^n); \quad n = 1, 2, \dots, N \quad (86)$$

Comparing equation (78) with equation (33), it is seen that

$$G(x_1^{n-1}; \theta^n) = \theta^n \quad (87)$$

and therefore that:

$$\frac{\partial G(x_1^{n-1}; \theta^n)}{\partial \theta^n} = 1 \quad (88)$$

$$\frac{\partial G(x_1^{n-1}; \theta^n)}{\partial x_1^n} = 0 \quad (89)$$

It is seen by comparison that equation (85) can be obtained from equation (41) by substituting θ^n for v^n and $-R(x_1^n)$ for u^n .

Accordingly, both $\frac{\partial F(x_1^{n-1}; \theta^n)}{\partial \theta^n}$ and $\frac{\partial F(x_1^{n-1}; \theta^n)}{\partial x_1^{n-1}}$ can be procured

from equations (52) and (53) by the same substitutions, thus:

$$\frac{\partial F(x_1^{n-1}; \theta^n)}{\partial \theta^n} = \frac{R(x_1^n)}{1 - \theta^n \frac{dR(x_1^n)}{dx_1^n}} \quad (90)$$

and

$$\frac{\partial F(x_1^{n-1}; \theta^n)}{\partial x_1^{n-1}} = \frac{1}{1 - \theta^n \frac{dR(x_1^n)}{dx_1^n}} \quad (91)$$

Inserting equations (88) through (91) into equation (39) yields:

$$\frac{1 - \theta^n \frac{dR(x_1^n)}{dx_1^n}}{R(x_1^n)} = \frac{1}{R(x_1^{n+1})}; \quad n = 1, 2, \dots, N \quad (92)$$

Combining equations (85) and (92) gives:

$$x_1^{n-1} = x_1^n + \frac{R(x_1^n) \left[\frac{R(x_1^n)}{R(x_1^{n+1})} - 1 \right]}{\frac{dR(x_1^n)}{dx_1^n}} \quad (93)$$

As before, the optimal values of x_1^n , $n = 1, 2, \dots, N$ may be found by the repetitive uses of equation (93), together with the initial and final conditions given by equation (80). The optimal holding time for each reactor may be recovered from equation (85).

Suppose that the rate of reaction can be written as:

$$r(x_1; T) = k_1 (T)^{\frac{s}{1}} (c_{i0} + \alpha_i x_1)^{\beta_i} - k_2 (T)^{\frac{s}{1}} (c_{i0} + \alpha_i x_1)^{\gamma_i} \quad (94)$$

where β_1, \dots, β_s and $\gamma_1, \dots, \gamma_s$ are exponents denoting the order of the forward and backward reactions with respect to each species. It will be further assumed that $\gamma_i - \beta_i = \alpha_i$, which is true for elementary reactions. k_1 and k_2 are, according to Arrhenious' law, functions of T only and have the form

$$k_i = k_{i0} \exp \left[-E_i/RT \right]; \quad i = 1, 2 \quad (95)$$

Applying equation (84) to equation (94) gives:

$$\frac{\partial r(x_1; T)}{\partial T} = \frac{1}{RT^2} \left[E_1 k_1 \frac{s}{1} (c_{i0} + \alpha_i x_1)^{\beta_i} - E_2 k_2 \frac{s}{1} (c_{i0} + \alpha_i x_1)^{\gamma_i} \right] = 0 \quad (96)$$

which can be reduced to:

$$\frac{k_1(T_m)}{k_2(T_m)} = K(T_m) = \frac{E_2}{E_1} \frac{s}{1} (c_{i0} + \alpha_1 x_1)^{\alpha_1} \quad (97)$$

Here T_m represents the temperature which maximizes the rate of reaction for exothermic reaction.

An explicit expression for T_m may be obtained by letting $(-\Delta H) = E_2 - E_1$, and substituting equation (95) into equation (97):

$$T_m = \frac{(-\Delta H)}{R} \left[\ln \frac{k_{20} E_2}{k_{10} E_1} \frac{s}{1} (c_{i0} + \alpha_1 x_1)^{\alpha_1} \right]^{-1} \quad (98)$$

If there is restriction on the temperature of the form $T_* \leq T \leq T^*$, then $T_m = T_*$ when the value of T_m given by equation (98) lies below T_* , and $T_m = T^*$ when it lies above T^* .

When T_m lies in the range (T_*, T^*) , substituting equation (98) into equation (94) gives:

$$\begin{aligned} R(x_1) = k_{10} \exp \left[\frac{-E_1}{(-\Delta H)} \ln \frac{k_{20} E_2}{k_{10} E_1} \frac{s}{1} (c_{i0} + \alpha_1 x_1)^{\alpha_1} \right] \\ \cdot \frac{s}{1} (c_{i0} + \alpha_1 x_1)^{\beta_1} - k_{20} \exp \left[\frac{-E_2}{(-\Delta H)} \ln \frac{k_{20} E_2}{k_{10} E_1} \frac{s}{1} (c_{i0} + \alpha_1 x_1)^{\alpha_1} \right] \\ \cdot \frac{s}{1} (c_{i0} + \alpha_1 x_1)^{\gamma_1} \end{aligned} \quad (99)$$

Letting $p = E_1/(-\Delta H)$, equation (99) can be simplified to:

$$\begin{aligned}
 R(x_1) &= k_{10} \left[\frac{k_{20} E_2}{k_{10} E_1} \frac{s}{1} (c_{i0} + \alpha_i x_1)^{\alpha_i} \right]^{-p} \cdot \frac{s}{1} (c_{i0} + \alpha_i x_1)^{\beta_i} \\
 &\quad - k_{20} \left[\frac{k_{20} E_2}{k_{10} E_1} \frac{s}{1} (c_{i0} + \alpha_i x_1)^{\alpha_i} \right]^{-(p+1)} \cdot \frac{s}{1} (c_{i0} + \alpha_i x_1)^{\gamma_i} \\
 &= \frac{s}{1} (c_{i0} + \alpha_i x_1)^{\beta_i - p \alpha_i} \frac{(k_{10})^{p+1}}{(k_{20})^p} \left[\left(\frac{E_1}{E_2} \right)^p - \left(\frac{E_1}{E_2} \right)^{p+1} \right]
 \end{aligned}$$

which finally reduces to:

$$R(x_1) = \frac{(p)^p}{(p+1)^{p+1}} \frac{(k_{10})^{p+1}}{(k_{20})^p} \frac{s}{1} (c_{i0} + \alpha_i x_1)^{\beta_i - p \alpha_i} \quad (100)$$

In case of a first-order reversible reaction $A_1 - A_2 = 0$, with $c_{10}/c_0 = 0$ and $c_{20}/c_0 = 1$, equation (100) becomes:

$$R(x_1) = c_0 Q P \left(\frac{x_1}{c_0} \right) \quad (101)$$

where $c_0 = c_{10} + c_{20}$; $Q = \frac{(p)^p}{(p+1)^{p+1}} \cdot \frac{(k_{10})^{p+1}}{(k_{20})^p}$; and $P \left(\frac{x_1}{c_0} \right) =$

$\left(\frac{x_1}{c_0} \right)^{-p} \left(1 - \frac{x_1}{c_0} \right)^{1+p}$. Letting $\xi = \frac{x_1}{c_0}$, equation (101) becomes:

$$R(x_1) = c_0 Q (\xi)^{-p} (1 - \xi)^{1+p} \quad (102)$$

Substituting equation (102) into equation (93) gives:

$$\xi^{n-1} = \xi^n - \frac{\xi^n (1 - \xi^n)}{\xi^n + p} \left[\left(\frac{\xi^{n+1}}{\xi^n} \right)^p \left(\frac{1 - \xi^n}{1 - \xi^{n+1}} \right)^{p+1} - 1 \right]$$

$$n = 1, 2, \dots, N-1 \quad (103)$$

The use of this recursion relation is illustrated by the following numerical example.

It is desired to find the optimal temperatures and holding times to increase c_1 from 0 g. mole/c.c. to 0.8 g. mole/c.c. in a three-tank reactor sequence for a first-order reversible reaction $A_1 - A_2 = 0$, with the following kinetic data:

$E_1 = 9.2$	Kcal/g. mole
$E_2 = 12.5$	Kcal/g. mole
$k_{10} = 10^{5.4}$	Min ⁻¹
$k_{20} = 10^{7.3}$	Min ⁻¹
$c_{20} = 1$	g. mole/c.c.

The values for p and Q are computed from the given data as

$$p = 2.79$$

$$Q = 0.1418 \quad \text{Min}^{-1}$$

By the iterative applications of equation (103), and a few trial calculations, the optimal values for ξ^n are obtained as

$$\xi^0 = 0$$

$$\xi^1 = 0.5945$$

$$\xi^2 = 0.7365;$$

hence:

$$c_1^0 = 0 \quad \text{g. mole/c.c.}$$

$$c_1^1 = 0.5945 \quad \text{g. mole/c.c.}$$

$$c_1^2 = 0.7365 \quad \text{g. mole/c.c.}$$

$$c_1^3 = 0.8 \quad \text{g. mole/c.c.}$$

Substituting the values of c_0 , Q , and p into equation (101) gives:

$$R(x_1^n) = \frac{0.1418 \cdot (1 - \xi^n)^{3.79}}{(\xi^n)^{2.79}} \text{ g. mole/c.c.} \cdot \text{min.}$$

From equation (85), the holding time can be expressed as

$$\theta^n = \frac{x_1^n - x_1^{n-1}}{R(x_1^n)}$$

If $n = 1$, the following is obtained:

$$\theta^1 = \frac{x_1^1 - x_1^0}{R(x_1^1)} = \frac{x_1^1 - x_1^0}{0.1418} \cdot \frac{(\xi^n)^{2.79}}{(1 - \xi^n)^{3.79}} = 29.9 \text{ min.};$$

Similarly, $\theta^2 = 66.8 \text{ min.}$, and $\theta^3 = 106 \text{ min.}$ For this particular case, equation (98) reduces to

$$T^n = \frac{E_2 - E_1}{R \ln \left(\frac{p+1}{p} \cdot \frac{k_{20}}{k_{10}} \cdot \frac{\xi^n}{1-\xi^n} \right)} = \frac{1661}{\ln \frac{107.9 \xi^n}{1 - \xi^n}}$$

and consequently:

$$T^1 = 328.0 \text{ K}$$

$$T^2 = 291.5^\circ \text{ K}$$

$$T^3 = 274.1^\circ \text{ K}$$

APPLICATION TO HETEROGENEOUS SYSTEMS

A heterogeneous system is defined as a system in which there are two or more different performance equations. It has been mentioned previously that optimization of such a system can also be accomplished by the direct application of the basic algorithm, equation (24). As an illustration, the optimal design for a multi-stage chemical process with parallel redundancy will be determined by the use of this algorithm.

Fig. 4 shows a multi-stage system in which a primary raw material is reacted with a secondary specie in the first stage to produce an intermediate product which is then fed to the second stage and reacted with another secondary specie and so on through the entire system.

Suppose that the secondary species are all quite unstable and cannot be stored and therefore must be produced upon demand by special reactions. All the intermediate products of the reactions are also assumed to be unstable. Then it is clear that, if a secondary specie is not available on time at any stage, the entire processing system will fail.

Failure is a stochastic phenomenon and therefore can be considered from a probabilistic point of view. The probability that the n^{th} secondary specie will be available on time is called the reliability of the n^{th} stage and represented by R^n . The reliability of the whole system, R_g , is the probability that all N secondary species are available on time (6), (17); thus:

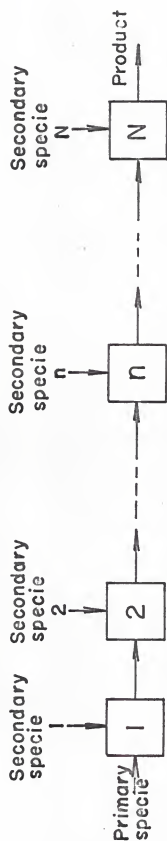


Fig.4. Series processing sequence.

$$R_s = \prod_{n=1}^N R^n \quad (104)$$

If the process which produces a certain secondary specie fails frequently, it would be advisable to produce more than one batch of that specie to increase the probability that it will be available on time. The production of more than one batch to reduce the effects of failures is named paralld redundancy.

Suppose that b^n batches of the n^{th} secondary specie are prepared. Since only one batch is needed, $b^n - 1$ batches are redundant. The probability that a single batch will fail is $1 - R^n$. Thus, the probability that all b^n batches will fail is equal to $(1 - R^n)^{b^n}$. Hence, the probability that at least one batch will succeed is $1 - (1 - R^n)^{b^n}$ which is by definition the reliability of the n^{th} stage with its redundancies. Thus, the reliability of the entire system can be represented by

$$R_s = \prod_{n=1}^N \left[1 - (1 - R^n)^{b^n} \right] \quad (105)$$

Since $b^n \geq 1$ and $R^n \leq 1$, it can be shown that

$$1 - (1 - R^n)^{b^n} \geq R^n$$

which indicates that the reliability of the system is increased by the use of parallel redundancy.

Let P_g be the profit obtained if the performance of the entire system is successful. The expected profit for the system is then expressed by $P_g R_s$. Let C^n be the construction cost of one reactor for the production of the n^{th} secondary specie,

(the cost is properly distributed over the life of the process), and O^n be the operation cost. Then, the net profit for the entire system is:

$$P = P_g R_s - \sum_{n=1}^N (C^n + O^n) b^n \quad (106)$$

The optimal parallel redundancy is the design which maximizes the net profit. It is to be noted that the expression for the net profit actually must include the construction and operation costs of the reactors in the main line for the production of the intermediate and final products, as well as the cost of the raw material entering the first stage. However, these costs are independent of the number of redundant reactors and hence it can be seen that the optimal design of parallel redundancy can be based on the net profit expressed by equation (106), considering only those costs associated with redundancy.

Rudd (6) has applied the method of dynamic programming in determining the optimal redundancy. Although dynamic programming is used for reducing the computational labor, it is felt that construction of the dynamic programming table at each stage is still a tedious task. The maximum principle is found to be more suitable for this particular problem.

To formulate this problem in the standard form, two state variables are introduced, satisfying the following relations:

$$x_1^n = x_1^{n-1} [1 - (1 - R^n) b^n]; \quad n = 1, 2, \dots, N \quad (107)$$

$$x_2^n = x_2^{n-1} - (c^n + o^n) b^n; \quad n = 1, 2, \dots, N \quad (108)$$

$$x_2^N = x_2^{N-1} - (c^N + o^N) b^N + P_g x_1^{N-1} [1 - (1 - R^N) b^N] \quad (109)$$

$$x_1^0 = 1 \quad (110)$$

$$x_2^0 = 0 \quad (111)$$

It should be noted that the values of R^n , c^n , and o^n are different for different n ; hence, the performance equations are different from stage to stage. It should also be noted that the performance equation at the N^{th} stage for x_2 is quite different from those at the other stages. Thus the process represented by these performance equations is a heterogeneous system.

It can be seen that x_1^n is the reliability of the sub-system consisting of the n upstream stages; and that x_2^N is equivalent to the net profit represented by equation (106). Consequently, the problem is now transformed into one in which x_2^N is to be maximized by the proper choices of b^n , $n = 1, 2, \dots, N$.

According to the basic algorithm, equation (24), z_1^n and z_2^n are defined as:

$$z_1^{n-1} = [1 - (1 - R^n) b^n] z_1^n; \quad n = 1, 2, \dots, N-1 \quad (112)$$

$$z_1^{N-1} = [1 - (1 - R^N) b^N] z_1^N + P_g [1 - (1 - R^N) b^N] z_2^N \quad (113)$$

$$z_2^{n-1} = z_2^n; \quad n = 1, 2, \dots, N \quad (114)$$

Since $z_1^N = 0$, $z_2^N = 1$, equations (113) and (114) become:

$$z_1^{N-1} = P_g \left\{ 1 - (1 - R^N) b^N \right\} \quad (115)$$

$$z_2^n = 1; \quad n = 0, 1, \dots, N \quad (116)$$

The Hamiltonians are:

$$H^n = z_1^n x_1^{n-1} \left\{ 1 - (1 - R^n) b^n \right\} + x_2^{n-1} - (c^n + o^n) b^n$$

$$n = 1, 2, \dots, N-1$$

and

$$H^N = x_2^{N-1} - (c^N + o^N) b^N + P_g x_1^{N-1} \left\{ 1 - (1 - R^N) b^N \right\}$$

The maximum may be found where

$$\frac{\partial H^n}{\partial b^n} = - z_1^n x_1^{n-1} (1 - R^n) b^n \ln (1 - R^n) - (c^n + o^n) = 0 \quad (117)$$

and

$$\frac{\partial H^N}{\partial b^N} = - (c^N + o^N) - P_g x_1^{N-1} (1 - R^N) b^N \ln (1 - R^N) = 0 \quad (118)$$

The solutions of equations (117) and (118) for z_1^n and P_g are, respectively:

$$z_1^n = - \frac{c^n + o^n}{x_1^{n-1} (1 - R^n) b^n \ln (1 - R^n)}; \quad n = 1, 2, \dots, N-1 \quad (119)$$

$$P_E = - \frac{C^N + O^N}{x_1^{N-1} (1 - R^N)^b \ln (1 - R^N)} \quad (120)$$

Combination of equations (107), (112), and (119) gives:

$$x_1^{n-2} = x_1^{n-1} \left[1 + \frac{(C^{n-1} + O^{n-1}) \ln (1 - R^n)}{(C^n + O^n) \ln (1 - R^{n-1})} \left(\frac{x_1^{n-1}}{x_1^n} - 1 \right) \right] \quad (121)$$

$$n = 2, 3, \dots, N-1$$

Combining equations (107), (115), and (120) yields:

$$x_1^{N-2} = x_1^{N-1} \left[1 + \frac{(C^{N-1} + O^{N-1}) \ln (1 - R^N)}{(C^N + O^N) \ln (1 - R^{N-1})} \left(\frac{x_1^{N-1}}{x_1^N} - 1 \right) \right] \quad (122)$$

It can be seen that equation (122) may be generated from equation (121) by letting $n = N$.

The optimal design for parallel redundancy then can be determined by the following procedure.

By assigning a value to x_1^N , the values for x_1^{N-1} and b^N can be obtained by solving equations (107) and (120) simultaneously, and the corresponding value for x_1^0 can eventually be calculated by iterative applications of equation (121). The procedure is repeated until the calculated value for x_1^0 is equal to 1. The values of b^n are then recovered from equation (107). As in the case of cross-current extraction without recycle, the computed values of x_1^n , $n = 1, 2, \dots, N$ are the optimal state variables corresponding to the value of x_1^0 , obtained in each run of trial

calculations. It should be noted that x_1^0 represents the reliability that the raw material will be available on time and hence can be any number between zero and one.

The same numerical example illustrated in Reference (6) is recalculated by the procedure described. The process under consideration consists of three stages, in which reactions proceed as follows:



The profit associated with the final product Z is $P_g = 10$.

The reliabilities and costs of operation are:

	$\underline{R^n}$	$\underline{C^n}$	$\underline{O^n}$
Stage 1	1/3	0.1	0.1
Stage 2	1/2	0.5	0.5
Stage 3	3/4	0.5	0.5

The results of calculation, after being rounded off, are $b^1 = 7$, $b^2 = 3$, and $b^3 = 2$, which are exactly the same as those obtained by the use of the dynamic programming algorithm.

ACKNOWLEDGEMENT

The author wishes to express his sincere appreciation to Dr. Liang-tseng Fan, for his constant enthusiasm and advice during the process of this work; Dr. William H. Honstead, Head of the Department of Chemical Engineering, for his help and encouragement; Dr. Benjamin G. Kyle and Dr. William L. Stamey, for reading the manuscript; and the K. S. U. Engineering Experiment Station for supporting this work (Project 345 supervised by Dr. Liang-tseng Fan).

BIBLIOGRAPHY

1. Bellman, R., "Dynamic Programming," Princeton Univ. Press, New Jersey, 1957.
2. Aris, R., "The Optimal Design of Chemical Reactors," Academic Press, New York, 1961.
3. Aris, R., Rudd, D. F., and Amundson, N. R., Chem. Eng. Sci., 12, 88, (1960).
4. Roberts, S. M., "Dynamic Programming Formulation of the Catalyst Replacement Problem," A paper presented at the Symposium on Optimization in Chemical Engineering, New York University, New York, (May, 1960).
5. Roberts, S. M., and Mahoney, J. D., Chem. Eng. Prog. Symp. Series, 58, 37, (1962).
6. Rudd, D. F., I. & E. C. Fundamentals, 1, 138, (1962).
7. Rudd, D. F., Chem. Eng. Sci., 17, 609, (1962).
8. Dranoff, J. S., Mitten, L. G., Stevens, W. F., and Wanninger, L. A., Jr., Operations Research, 9, 388, (1961).
9. Mitten, L. G., and Nemhauser, G. L., "Multistage Optimization with Dynamic Programming," A paper presented at the Los Angeles Meeting of A. I. Ch. E., (February, 1962).
10. Katz, S., I. & E. C. Fundamentals, 1, 226, (1962).
11. Katz, S., J. of Electronics and Control, 13, 179, (1962).
12. Katz, S., Private communication, (December, 1962).
13. Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F., "The Mathematical Theory of Optimal Processes," (English translation by Tiriogoff, K. N.),

Interscience Publishers, New York, 1962.

14. Converse, A. O., "Computer Optimization of a Multistage Allocation Problem by Means of a Non-Imbedding Technique," A paper presented at the Chicago Meeting of A. I. Ch. E., (December, 1962).
15. Rudd, D. F., and Blum, E. D., Chem. Eng. Sci., 17, 277, (1962).
16. Jackson, R., Chem. Eng. Sci., 18, 215, (1963).
17. Bazovsky, I., "Reliability Theory and Practice," Prentice-Hall, Inc., New Jersey, 1961.

NOMENCLATURE

a	= a given value of the state variable.
A	= a constant in equation (62).
A_i	= the i^{th} chemical specie.
b	= number of batches.
B	= a constant in equation (62).
c	= concentration.
c_o	= $c_{1o} + c_{2o}$
C	= construction cost.
E	= activation energy.
H	= the Hamiltonian.
ΔH	= $E_1 - E_2$.
k	= reaction rate constant.
K	= k_1/k_2 .
N	= total number of stages.
O	= operation cost.
p	= $E_1/(-\Delta H)$.
P	= net profit.
P_g	= gross profit.
q	= flow rate of the feed.
Q	= $\frac{(p)^p}{(p+1)^{p+1}} \cdot \frac{(k_{10})^{p+1}}{(k_{20})^p}$
r	= flow rate of the recycle stream.
$r(x; T)$	= reaction rate.
R	= the ideal gas law constant.

R_s	= reliability of the entire system.
R^n	= reliability of the n^{th} stage.
$R(x)$	= the reaction rate at $T = T_m$.
S	= total number of state variables.
t	= total number of control variables.
T	= temperature.
T_m	= the optimal temperature.
u	= concentration of solute in extract.
v	= $w/(q + r)$.
w	= flow rate of wash water.
W	= raw material.
x	= state variable.
\bar{x}	= the optimal value of x .
X	= an intermediate product.
y	= perturbation of the state variable.
Y	= an intermediate product.
z	= a variable introduced in the basic algorithm, equation (24).
Z	= final product.

Greek Letters

α	= stoichiometric coefficient.
β	= order of the forward reaction.
γ	= order of the backward reaction.
δ_{im}	= the Kronecker delta.
ϵ	= a small number.
ζ	= the lower bound of the control variable.

- η = the upper bound of the control variable.
- u = control variable.
- $\bar{\theta}$ = the optimal value of θ .
- λ = relative cost of wash water.
- μ = constant in linear phase equilibrium relation.
- ξ = x/c_0
- φ = perturbation of the control variable.

A STUDY OF THE OPTIMIZATION
OF SOME CHEMICAL PROCESSES

by

CHIU SEN WANG

B.S., National Taiwan University, 1960

AN ABSTRACT OF A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Chemical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1963

The major purpose of this work was to develop a method of optimization particularly suitable for the design of multi-stage chemical processes. A basic algorithm for optimizing multi-stage heterogeneous systems with product recycle was derived as an extension of the maximum principle, which was originally proposed by Pontryagin for continuous processes, to the discrete systems.

It was found that the discrete version of the maximum principle offers many computational advantages over the method of dynamic programming, because it avoids the use of interpolation which usually leads to an immense computer memory requirement and considerable interpolation error. The discrete version of the maximum principle proved to be a very convenient and powerful tool in many cases, especially for the systems with product recycle.

The general working equations and the computational scheme are presented for a special class of one-dimensional systems, to which many industrial processes belong. The working equations, consisting of a recurrent relation and a final condition, can be easily applied to practical problems. To illustrate the use of these equations, the following two problems were worked out in detail.

(1) Cross-current extraction process: The working equations were first derived for the system with product recycle, and then reduced to the case without recycle. Generally, a numerical method must be used to obtain the final solutions for the process with a non-linear phase equilibrium relation. For

the simple case of a linear phase equilibrium relation, analytical solution is obtainable.

(2) The continuous flow stirred tank reactor with a single reaction: This is a system with two control variables -- temperature and holding time. The well-known disjoint characteristic of the optimal temperature policy was proved mathematically by means of the basic algorithm. A recurrent relation was derived for the calculations of the optimal holding times.

The application of the basic algorithm to the optimization of heterogeneous systems is also illustrated by the design of parallel redundancies in chemical processes.